基于多元关系的张量分解标签推荐方法*

曾　辉，胡　强，淦修修

(华东交通大学 信息工程学院，南昌 330013)

摘　要：标签推荐广泛应用于各大网站，如电影网站，电子商务网站等等，但现有方法忽视了多种属性特征之间的联系，无法保证大数据环境下推荐系统的准确率。针对该问题，提出了一种基于用户聚类和张量分解的新标签推荐方法，以进一步提高标签推荐的质量。该方法首先对一些对产品具有重要影响的用户进行聚类，然后根据用户、产品、标签和产品评分之间的多元关系综合计算总权重。最后，根据聚类之后的用户群体以及多元关系的总权值构建张量并进行张量因式分解。与传统张量分解方法相对比，实验结果表明本文提出的方法在准确率上具有一定的提高，验证了算法的有效性。

关键词：标签推荐；张量因子分解；权重；聚类

中图分类号：TP181　　doi: 10.3969/j.issn.1001-3695.2018.04.0215

Method for tag recommendation of tensor decomposition based on multiple relationships

Zeng Hui, Hu Qiang, Gan Xiuxiu

(*College of Information Engineering*, *East China Jiaotong University*, *NanChang* 330013, *China*)

Abstract: Nowadays, tag recommendation is widely used in various websites, such as movie websites, e-commerce websites and so on. However, there are some methods that ignore the connection among the characteristics of a variety of attributes and can not guarantee the accuracy of the recommender system in the big data environment. Aiming at this problem, this paper proposed a tag recommendation method based on user clustering and tensor decomposition, which could further improve the quality of tag recommendation. The method firstly clustered the users who had an important influence on the product, and then comprehensively calculated the total weight based on the multiple relationships among the users, products, tags, and product ratings. Finally, it constructed the tensor according to the user groups after clustering and the total weight of the multivariate relations, and performed the tensor factorization. Compared with the traditional tensor decomposition method, the experimental results show that our method improves the accuracy and verifies the effectiveness of the algorithm.

Key words: tag recommendation; tensor factorization; weight; clustering

通常，标签推荐意味着不同的用户可以使用不同的标签（字词列表）来标注产品（如网站、电影），然后根据用户过去的标记行为预测其未来的行为，向目标用户推荐其感兴趣的产品。例如，若两个不同的用户都标记了相同的商品，则他们趋向于使用相同的标签注释未来的其他商品。

目前，部分标签推荐方法通过张量分解技术对标签进行排序，例如高阶奇异值分解（HOSVD）[1]，张量因子分解排序算法（RTF）[2]和用于情境感知协同过滤的 n 维张量因子分解[3]等等。Rendle 等人[2]引入了两种不同的张量数据解释：0/1 解释方案和基于位置的排序解释方案，结果显示 RTF 能够获得良好的预测质量。杨秋勇[4]提出了一种基于三元关系权值数据解释方案（LORTF），虽然实验结果有一定的提高，但是算法效率较低。在本文中，提出了一种通过考虑用户、产品、标签、评分之间多元关系计算综合权重，从而构造出基于四元关系的张量模型，实验结果表明本文提出的模型优于其他基准算法。

本文首先根据用户的活跃性以及相似性的差异对用户进行聚类，将数据集划分成若干个小数据集；其次，根据用户、产品、标签、评分之间的相互关系计算出各自的重要性，综合四元关系的权值构造三维张量模型；最后，应用 Tucker 分解法及最小二乘法对张量模型进行优化求解，生成一系列的推荐结果。实验结果表明，本算法的推荐准确率与其他推荐算法相比具有较好的提高，验证了本文算法的有效性。

0　相关工作

标签推荐已经被广泛地应用于各种模型推荐算法中。Krestel 等人[5]提出了一种基于潜在狄利克雷分配（LDA）的方

收稿日期：2018-04-03；**修回日期：**2018-05-12　　**基金项目：**国家自然科学基金资助项目（61562027）；江西省教育厅科学技术研究资助项目（GJJ170379）

作者简介：曾辉（1973-），男，江西赣州人，副教授，硕导，硕士，主要研究方向为软件工程、数据挖掘（macrohui29@sina.com）；胡强（1993-），男，江西吉安人，硕士研究生，主要研究方向为数据挖掘；淦修修（1994-），男，江西九江人，硕士研究生，主要研究方向为数据挖掘.

法，它应用具有潜在主题的专业术语向用户推荐相应的文章，同一主题中的词语则有机会在其他文章中呈现，但是该方法缺乏个性化的考虑。

随着推荐算法的快速发展，许多算法模型都被混合应用于推荐领域中，如贝叶斯分类器、聚类、人工神经网络以及决策树等等机器学习算法。文献[6]将社交标签并入两种聚类方法：基于 LDA 的 K-means 和生成聚类方法，尽管文中证明了标签作为聚类的附加信息源的价值，但是其中的聚类模型缺乏用户维度，导致 F1 得分略低。

张量因式分解模型已在各种个领域研究应用广泛[7~9]。Kolda 等人[10]描述了 CANDECOMP / PARAFAC 和 Tucker 分解的两种不同的张量因子分解方法。Frolov 等人[11]介绍了张量在社会标记系统中的应用以及各种相关的张量分解算法。

1　标签推荐

标签推荐的任务是为用户提供特定产品标签的个性化列表。例如，当观众（用户）想要标记电影（产品）时，电影网站能够从该用户的过去对其他标签的注释行为以及其他用户对该标签和其他标签的注释行为中学习推荐标签的列表，从而向观众推荐他或她想要的关键字列表。

1.1　数据的形式化表示

假设分别用 U、P、T、R 表示用户、产品、标签、评分集合。在文章[8]中将历史标记信息表示为 $A \subseteq U \times P \times T$，它是一个关于分类变量的三元关系，可以被看作是一个三维张量，其中的 A 中的三元组是过去的积极观测值。本文中的元组使用的是“用户-产品-标签-评分”组成的元组，但评分并不作为主要的特征维度，只是作为一个重要的属性特征，应用在后面的权值计算中。元组 $(u,p,t,s) \in A$ 表示存在用户 u 对产品 p 使用标签 t 进行了标识并给出相应的评分 s，表明用户 u 对所用的产品的描述或评价。分别结合 (u,p)，(u,t)，(p,t) 可以定义成 P_A, Q_A, R_A 的集合，且满足以下关系：

$$
\begin{aligned}
P_A &:= \{(u,p) \mid \exists t \in T : (u,p,t) \in A\} \\
Q_A &:= \{(u,t) \mid \exists p \in P : (u,p,t) \in A\} \\
R_A &:= \{(p,t) \mid \exists u \in U : (u,p,t) \in A\}
\end{aligned}
\tag{1}
$$

其中：P_A, Q_A, R_A 可以认为是元组 A 分别在用户-产品、用户-标签、产品-标签的维度上的二维投影。

1.2　张量数据的表示

张量具有不同的表示方式，每种表示方式能够应用在不同推荐算法中，以下主要介绍三种解释形式。

1.2.1 基于 0/1 解释方案

Y 中的三元组可以用三阶张量表示，Symeonidis 等人[1]提出将 Y 解释为一个稀疏张量，其中 1 表示正反馈，0 表示缺失值（如图 1 所示），训练数据 $y^{0/1}$ 被定义为

$$
y^{0/1}_{u,i,t} = \begin{cases} 1, & (u,i,t) \in A \\ 0, & \text{else} \end{cases}. \tag{2}
$$

但是，基于 0/1 解释方案具有语义错误、精确度较低的缺陷[2]。

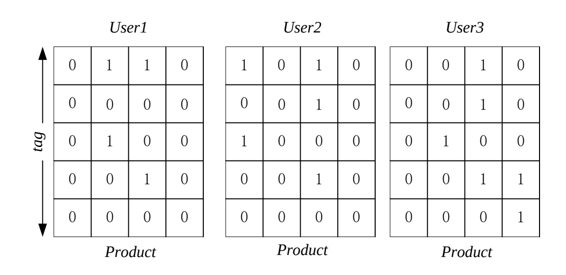


图 1　基于 0/1 解释表现形式图例

1.2.2 基于标注的排名解释方案

Rendle 等人[2]提出区分正、负示例和缺失值，以便学习标签的个性化排名。该原理是根据观察到的标签分布中积极、消极的例子判断正负，可观察到的标签被解释为正反馈，而未标记标签则为负反馈，剩余其他条目被假定为缺失值（如图 2 所示）。它可以根据以下形式定义积极和消极集合：

$$
\begin{aligned}
T^{+}_{u,p} &:= \{ t \mid (u,p) \in P_S \wedge (u,p,t) \in A \} \\
T^{-}_{u,p} &:= \{ t \mid (u,p) \in P_S \wedge (u,p,t) \notin A \}
\end{aligned}
\tag{3}
$$

其中：负反馈 $T^{-}_{u,p}$ 并不代表为零，只是正反馈 $T^{+}_{u,p}$ 的值要比负反馈 $T^{-}_{u,p}$ 的值更高，更具有参照意义。

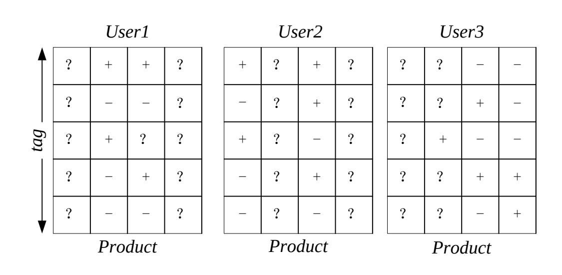


图 2　基于标注排序的表现形式图例

1.2.3 基于权值解释方案

根据“用户-产品-标签-评分”四元组之间的关系，将每个维度上的权重关系综合考虑，利用总权值 w_{u_j,i_k,t_l} 的方式表示该元组的重要性，从而表明了其在整个元组集合中所占的比重，其具体含义为用户 j 使用标签 l 来标记产品 k，表达了用户对产品的喜爱程度，如图 3 表示。

图 3　基于权值表现形式图例

基于综合权值的表现形式与前两种表现形式对比具有以下几个优点：

a）能够充分考虑元组中的每个元素之间的相互关系，并且元素之间各自的差异也可以加以区分，减少了相关属性对生成

推荐结果产生的不良影响。

b）能够充分考虑产品的受欢迎程度、用户对某些产品的喜欢程度，将使用过该产品的用户、标签等因素用权值表示，并将用户对该产品的评分等级进行加权计算，综合考虑之后产品所表现受欢迎程度层次分明，更易于精确推荐。

c）有助于区分标签的重要性。在同一产品中，每个用户所使用的标签可能不同，但可以将不同的标签进行统计筛选，将出现频率较高的标签给出相对较高权值，出现次数较少或者语义不明的标签权值相对较低。

2　方法

本章节首先介绍了张量因子分解模型，然后根据用户使用产品的活跃性以及用户之间相似性对用户聚类，在同一用户群体信息基础上找出相关的元组关系，基于权值的形式表示元组之间的重要性，结合“用户-产品-标签-评分”之间关系的综合权值构建张量模型，并对其进行优化求解，最后获得对目标用户的推荐列表。

2.1　张量因式分解模型

本文中，张量分解的方法主要是 Tucker 分解，是一种高阶主成分分析的形式。它在每种模式下将张量分解成核心张量以及对应的因子矩阵，分解原理如图 4 所示。

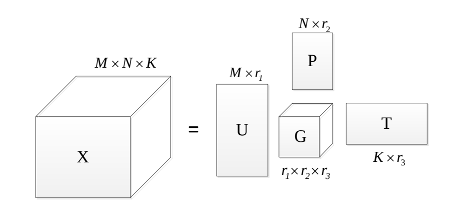


图 4　三阶张量的 Tucker 分解原理

张量 X 分解可以表示为：$X=[\![G;U,P,T]\!]\equiv G\times_1 U\times_2 P\times_3 T$。其中，$U\in\mathbb{R}^{M\times r_1}$，$P\in\mathbb{R}^{N\times r_2}$，$T\in\mathbb{R}^{K\times r_3}$ 是相应维度上的低秩特征矩阵，它可以被认为是每种模式的主要组成部分，因此被认为是一种高阶的主成分分析的形式。核心张量 $G\in\mathbb{R}^{r_1\times r_2\times r_3}$ 表示不同部分（特征矩阵）之间的交互，保留了原始张量主要信息并且具有一定的稳定性，它的维度为 $r_1\times r_2\times r_3$。在通常情况下，被压缩后的张量的存储量需求要远小于原张量，有利于满足存储空间的需求。在学习了特征矩阵和核心张量之后，可以按如下方式进行预测：

$$\hat{x}_{u,p,t}=\sum_{\tilde{u}}\sum_{\tilde{p}}\sum_{\tilde{t}}\hat{g}_{\tilde{u},\tilde{p},\tilde{t}}\cdot\hat{u}_{u,\tilde{u}}\cdot\hat{p}_{p,\tilde{p}}\cdot\hat{t}_{t,\tilde{t}}\,.\tag{4}$$

其中:波浪号表示特征矩阵中特征维度上的索引，用“帽子”标记特征矩阵的元素（如，$\hat{u}_{u,\tilde{u}}$）。能够从公式（4）中导出张量的预测值，并生成 Top-N 个性化推荐列表：

$$Top(u,p,N)=\underset{t\in T}{\overset{N}{\arg\max}}\,\hat{x}_{u,p,t}\,.\tag{5}$$

2.2　综合权重的计算方法

2.2.1 用户聚类

本文将所有用户集合根据它们之间的活跃性不同分为三大类：活跃用户，普通用户，无意用户。活跃用户是用户集合中最具代表性的类别，该类用户使用了大量产品并用标签标记、评分等，因此活跃用户自身的权重会比较大。相反，无意用户的权重将非常小，该类用户指在平台中极少的产品使用量，甚至注册之后仅仅上线过几次，没有过多的参考价值。除了活跃用户和无意用户之外，剩下的则为普通用户，该类用户占所有用户集合中的绝大多数，同时也是主要的推荐对象。

用户活跃性更多体现在用户使用的相关产品以及所标记的标签数量上，通常产品和标签的数据量通常要远远大于用户的数据量，因此在用户的权值计算时需要将产品和标签对用户的影响进行综合考虑。

首先计算用户使用产品的权值大小，计算公式如式（6）所示，其权值 w_{p_k} 的大小与使用该产品的用户次数成正比，能够反映该产品受到的欢迎程度。

$$w_{p_k}=\frac{\sum_{(u_k,p_k)\in P_A}u_k}{|U|}\tag{6}$$

但是由于每个用户使用的产品的数量都不一样，有些活跃用户使用的产品数量巨大，为了避免产品的使用量过多导致权值太大，需要设置一个权重因子 α_p 使得产品的总权值小于等于 1，计算形式如下所示：

$$w_{u_p}=\alpha_p*\sum_{(u_k,p_k)\in P_A}w_{p_k}\tag{7}$$

相似地，标签的权值大小与该标签被使用的用户数量成正比，其权值 w_{t_k} 可以使用式（8）计算。

$$w_{t_k}=\frac{\sum_{(u_k,t_k)\in Q_A}u_k}{|U|},\tag{8}$$

同理，由于某些标签被使用的次数过多，导致累加的权值过大，因此需要设置一个参数 α_t 限定标签的总权值小于等于 1，计算形式如下所示：

$$w_{u_t}=\alpha_t*\sum_{(u_k,t_k)\in Q_A}w_{t_k},\tag{9}$$

用户的权重需要通过使用的产品权重 w_{u_p} 和标签权重 w_{u_t} 综合计算，但是由于前者的重要性要偏高，并且需要限定用户权值在 0 和 1 之间，所以可以设置一个参数 α 且大于 0.5，计算形式如下：

$$w_u=\alpha*w_{u_p}+(1-\alpha)w_{u_t}\ (\alpha\in(0.5,1)).\tag{10}$$

获得所有用户的权重之后，依据所有用户权值大小进行降序排序，然后选择前 N 个用户作为活跃用户集，将排在最后部分且小于设定的阈值的用户集作为无意用户，该类用户不加入模型的训练，剩下的则为普通用户。

由于用户集合过大，若直接构建张量模型，则算法复杂性会明显增大，为了提升算法数据的处理效率以及提高推荐的准确性，将普通用户与活跃用户集合聚类成若干个相似性的用户群体。通过采用修正余弦相似度方法计算出用户之间的相似性，

将各个用户所使用的产品集合减去当前用户已评级产品的平均值作为输入向量，把活动用户 u_a 和普通用户 u_o 所使用的产品集合作为 m 维向量，其中向量元素值为集合中产品被用户使用的次数 ts。相似度计算方法如下：

$$sim(u_a,u_o)=\frac{\sum_{p_k\in(P_{u_a}\cap P_{u_o})}(ts_{u_a,p_k}-\overline{ts_{u_a}})(ts_{u_o,p_k}-\overline{ts_{u_o}})}{\sqrt{\sum_{p_k\in(P_{u_a}\cap P_{u_o})}(ts_{u_a,p_k}-\overline{ts_{u_a}})^2}\sqrt{\sum_{p_k\in(P_{u_a}\cap P_{u_o})}(ts_{u_o,p_k}-\overline{ts_{u_o}})^2}}. \quad (11)$$

根据计算的相似性结果对比分析，选择多个活跃用户作为聚类的中心目标，并对各自之间的相似性迭代排序，为每个普通用户选择一个最为相似的用户群体，最终获得多个聚类集。本文根据聚类结果筛选之后选取了两个聚类集作为实验数据，将聚类集中用户的相关产品、标签、评分构成元组数据，以便于下一步求解元组的综合权重。

2.2.2 计算元组的综合权重

计算综合总权值需要综合不同维度上的权值考虑，它的权值大小是元组关系的总体表现。用户权值体现的是用户的活跃程度，主要依据用户使用产品和标签的数量，而产品权值不仅要考虑用户在产品上的权重和标签在产品上的权重，而且还需要考虑用户对产品的评分大小。

用户在某个产品上的权值 $w_{p_k,u}$ 主要是依据使用该产品的所有用户权值 w_{u_l} 的平均值，可用式（12）表示。

$$w_{p_k,u}=\frac{\sum_{(u_l,p_k)\in P_A}w_{u_l}}{|U_{p_k}|} \quad (12)$$

标签在产品上的权重 $w_{p_k,t}$，指某个产品被所有标签标记的权值，通过某个产品被标签标记的数量和标签集合总数的比例计算可得：

$$w_{p_k,t}=\frac{\sum_{(p_k,t_l)\in R_A}t_l}{|T_{p_k}|} \quad (13)$$

评分权值 s_{p_k,u_s} 是指某个产品被大众用户给出的评分均值，可通过用户集合对该产品总评分与该产品被用户的评分次数的比例计算而得，可用式（14）表示。

$$s_{p_k,u_s}=\frac{\sum_{(p_k,u_s)\in P_A}u_s}{|U_{p_k}|} \quad (14)$$

产品的总权值 w_{p_k} 需要综合以上三者关系，但它们之间所占的重要关系不一样，需要设置不同的参数限定产品的总权值。并且由于产品的总权值最大值为 1，而其中的评分值普遍大于 1，因此需要在它们前面设置不同的限定参数 λ_i，计算方法如下：

$$w_{p_k}=\lambda_1\cdot w_{p_k,u}+\lambda_2\cdot w_{p_k,t}+\lambda_3\cdot s_{p_k,u_s} \quad (\lambda_i\in(0,1)) \quad (15)$$

标签的权值 w_{t_k} 的求解只需单纯地考虑标签之间的关系，无须再考虑用户和产品在标签上的影响因素，因为前面求得的用户权重和产品权重均将标签与它们之间的关系考虑进去。标签权值的求解是根据三元组中标签出现次数与所使用的标签的最大次数的比例计算：

$$w_{t_k}=\frac{\sum_{(u_k,p_k,t_k)\in A}times_{t_k}}{max(times_t)} \quad (16)$$

综合权重的计算可以根据以上所求得的用户权值 w_{u_k}、产品权值 w_{p_k} 和标签权值 w_{t_k} 综合考虑，并设定相应的比例参数 μ_i，计算公式如下所示：

$$w_{u_k,p_k,t_k}=\mu_1\cdot w_{u_k}+\mu_2\cdot w_{p_k}+\mu_3\cdot w_{t_k} \quad (\mu_i\in[0,1]) \quad (17)$$

将用户 U、产品 P、标签 T 分别作为张量的三个维度，由计算得到的综合权值确定对应张量维度上的元素值并构建三维张量。通过对张量进行 tucker 分解以及最小二乘法优化分解的结果，再根据式（4）（5）可以获得最终预测的 TOP-N 个性化推荐列表。

3 实验结果和分析

3.1 数据集

本文对初步获得的 MovieLens 数据集进行数据预处理之后，最终得到 16 3295 条记录。这些数据集中包含 500 个用户，9 289 个电影，1 128 个标签，每个用户都对自身看过的电影给出了评分，评分范围为 0.5-5 共十个等级，每个评分区间为 0.5。根据用户聚类的结果，将 500 个用户分成了六类，选取了聚类结果最好的两类用户的相关数据与其他算法进行对比实验，主要的实验数据如表 1 所示。

表 1 数据集说明

数据集	用户	项目	标签	总记录数
Cluster1	100	5196	1123	59524
Cluster2	84	4002	1108	28338

3.2 参数设置

在用户聚类的过程中，由于每个用户使用的产品数量以及标记的标签数量都不一样，甚至相差太大，导致用户使用的产品权值和标签权值累加数量过大，甚至有些用户使用的产品权值接近 100，因此参数会设置的相对较小，将参数 α_p 和参数 α_t 都设置为 0.01。在计算用户权值时，产品权值比重需要大于标签的权值，否则用户聚类效果会不理想，因此将参数 α 设置为 0.6。

在综合权值计算过程，其中参数 λ_1 为 0.01，参数 λ_2 为 0.01，参数 λ_3 为 0.05，参数 $\mu_1=\mu_2=\mu_3=1$。在计算产品的权值时，参数 λ_1 和参数 λ_2 两个参数的设置缘由与用户聚类的相类似，由于累加数量过多因此需要设置一个较小的参数，评分权值参数 λ_3 由于评分本身数值过大，产品权值又限定小于等于 1，因此也需要设定一个较小参数。最后的综合权值计算中，三个特征维度上的参数设置都是 1，即保证总权值小于等于 3。

3.3 评估方法

本文实验的评价指标是在各类算法研究中常被应用的准确率（precision）、召回率（recall）、F 值（F-measure）/（F-score）。

本算法从两个簇类集的用户中，随机抽取每个用户所使用的部分相关产品及标签数据作为测试集 S_{test}，剩余部分作为训练集 S_{train}。每个评价指标都是使用推荐列表中的 TOP-1 至 TOP-15 进行对比实验。

$$\text{Precision}(S_{test},\text{N})=\underset{(u,p)\in S_{test}}{\text{avg}}\frac{\left|\text{Top}(u,p,N)\cap\{t|(u,p,t)\in S_{test}\}\right|}{N} \tag{18}$$

$$\text{Recall}(S_{test},\text{N})=\underset{(u,p)\in S_{test}}{\text{avg}}\frac{\left|\text{Top}(u,p,N)\cap\{t|(u,p,t)\in S_{test}\}\right|}{\{t|(u,p,t)\in S_{test}\}} \tag{19}$$

$$\text{F1}(S_{test},\text{N})=\frac{2\cdot\text{Precision}(S_{test},\text{N})\cdot\text{Recall}(S_{test},\text{N})}{\text{Precision}(S_{test},\text{N})+\text{Recall}(S_{test},\text{N})}. \tag{20}$$

3.4 实验结果和分析

本文使用四个算法与本算法进行对比实验，其中“0/1 Scheme”是使用“0/1”模式[12]，“LORTF”虽然使用的也是权值的方式，但是该方法仅仅考虑的是三元关系[4]，而本方法“New Method”利用“用户-产品-标签-评分”四元关系权值模式进行的张量分解，更多地参考了四元关系之间的权重大小，“4-D”是采用四维张量分解的方法[11]，将“用户-产品-标签-评分”作为四个不同的维度构建张量进行对比。在这几种方法中，使用的都是相同的拆分方法和数据集，在每个数据集中随机抽取 20%的数据作为测试集，80%用作训练集构建模型进行实验。前四种算法的核张量都是使用（8，8，8）的维度，而算法“4-D”为了避免运算的复杂性过大和内存消耗过大，将评分等级改成 1-5 个等级，并使用（8，8，8，3）维度的核张量进行优化分解。以下应用不同的评价指标对两个簇类数据的对比实验结果进行分析。

对比图 6 和 7 中的实验结果可知，本方法在 Top-1 至 Top-15 推荐列表中计算的准确度优于其他方法，随着推荐数目地递增，算法的准确度也在逐渐地减小，并且可以看出 Cluster1 和 Cluster2 的实验结果精确度都超过了其他方法。

根据图 8 和 9 中的召回率对比可以看出，数据集 Cluster1 中 Top1 至 Top3 的结果优于其他方法，但是随着 Top-N 数量的递增，该方法的召回率略低于其他方法。通过与其他方法对比可知，在数据集 Cluster2 中，随着 Top-N 值的递增，本方法的召回率表现优于其他方法。

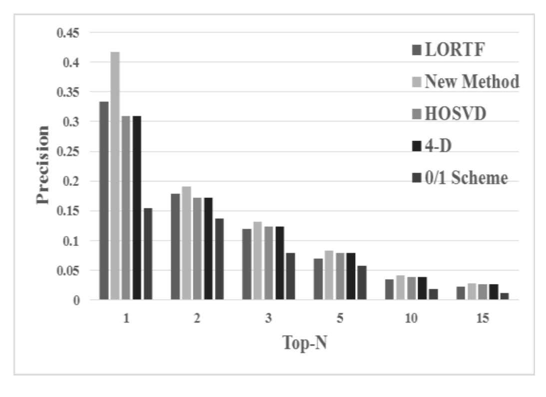


图 6　Cluster1 的 Top-1 到 Top-15 列表的准确率

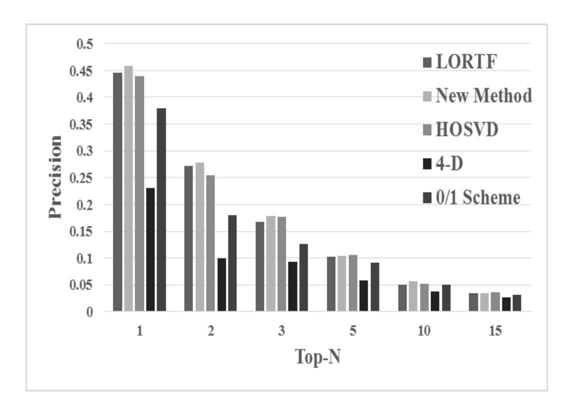


图 7　Cluster2 的 Top-1 到 Top-15 列表的准确率

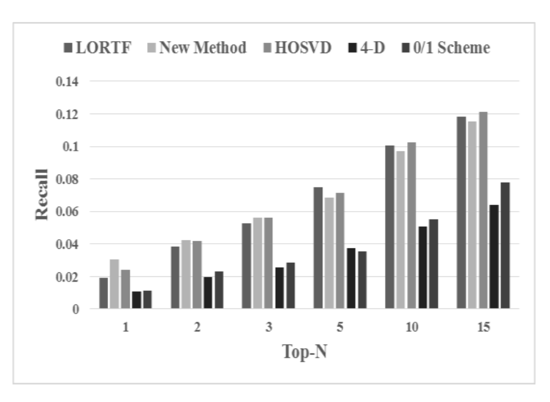


图 8　Cluster1 的 Top-1 到 Top-15 列表的召回率

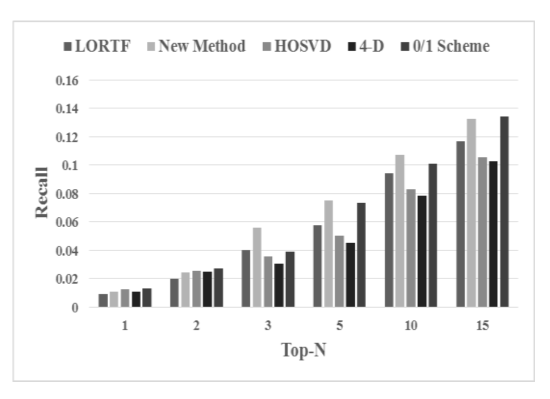


图 9　Cluster2 的 Top-1 到 Top-15 列表的召回率

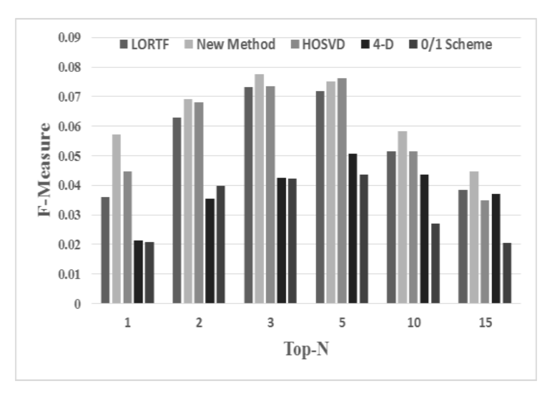


图 10　Cluster1 的 Top-1 到 Top-15 列表的 F-Measure

通过图 10 和图 11 中的各个方法对比，可以看出本文方法的 F-measure 值无论是在 Cluster1 还是在 Cluster2 中均要高于其他方法，并且推荐数量在 Top3-Top5 之间时明显高于其他 Top 值，但整体 F-Measure 很小，这是由于数据集太稀疏的缘故。结果表明，本方法在推荐过程中的表现明显好于其他方法。其原因主要是本方法充分考虑了多元关系的属性特征，以权值区

分各自的重要性，间接地提升了推荐的准确率。

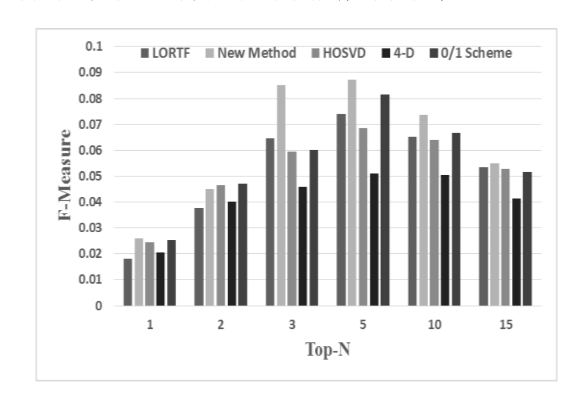


图 11 Cluster2 的 Top-1 到 Top-15 列表的 F-Measure

4 结束语

本文首先针对三种不同的张量数据表现形式进行了介绍，其中包括 0/1 解释方案，标注排序解释方案和综合权重解释方案。然后，基于用户的活跃性以及相似性对用户进行聚类，并在聚类后的用户集合基础上，结合产品、标签、评分的多元关系计算综合权值。将本文方法的评估结果与其他方法相对比，结果表明本文方法具有较好的准确性、合理性。对于未来的工作，将从下面两个方面进行研究：a）通过调节更优的参数，或者结合一些其他重要影响因素，提升权值的计算方法；b)研究一种更好的方法来应对数据的稀疏性，以提高算法的运算速度和预测的准确性。

参考文献：

[1] Symeonidis P. User recommendations based on tensor dimensionality reduction [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2008: 43-50.

[2] Rendle S, Marinho L B, Nanopoulos A, *et al.* Learning optimal ranking with tensor factorization for tag recommendation [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 727-736.

[3] Maroulis S, Boutsis I, Kalogeraki V. Context-aware point of interest recommendation using tensor factorization [C]// Proc of IEEE International Conference on Big Data. 2017: 963-968.

[4] 杨秋勇. 应用张量分解法学习最优评分的推荐系统研究 [D]. 广州: 华南理工大学, 2012. (Yang Qiuyong. Research of learning optimal ranking with tensor factorization for recommendation system [D]. Guangzhou: South China University of Technology, 2012)

[5] Krestel R, Fankhauser P, Nejdl W. Latent dirichlet allocation for tag recommendation [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2009: 61-68.

[6] Lu X S, Zhou M C. Analyzing the evolution of rare events via social media data and k-means clustering algorithm [C]// Proc of IEEE International Conference on Networking, Sensing, and Control. 2016: 1-6.

[7] Acar E, Dunlavy D M, Kolda T G. A scalable optimization approach for fitting canonical tensor decompositions [J]. Journal of Chemometrics, 2015, 25 (2): 67-86.

[8] Goulart J, Boizard M, Boyer R, *et al.* Tensor CP decomposition with structured factor matrices: algorithms and performance [J]. IEEE Journal of Selected Topics in Signal Processing, 2016, 10 (4): 757-769.

[9] Zhou G, Cichocki A, Zhao Q, *et al.* Nonnegative matrix and tensor factorizations: an algorithmic perspective [J] , IEEE Signal Processing Magazine, 2014, 31 (3): 54-65, .

[10] Kolda T G, Bader B W. Tensor decompositions and applications [J]. Society for Industrial and Applied Mathematics, 2009, 51 (3): 455-500, .

[11] Frolov E, Oseledets I. Tensor methods and recommender systems [J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2016.

[12] Symeonidis P, Nanopoulos A, Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction [C]// Proc of IFIP International Conference on Artificial Intelligence Applications and Innovations. Boston: Springer, 2009: 331-340.